

# DC Motor Velocity Control Using Pulse Width Modulation (PWM)

e-Yantra Team

Embedded Real-Time Systems Lab  
Indian Institute of Technology-Bombay

IIT Bombay  
March 9, 2020



# Agenda for Discussion

## 1 Introduction

- Pulse Width Modulation
- Duty Cycle

## 2 PWM Generation in AVR

- Timers in AVR
- Timer/Counter 5 (TCNT5)
- Output Compare Register
- TCCR5A
- TCCR5B
- Summary



# Pulse Width Modulation



# Pulse Width Modulation

- ① Pulse Width Modulation (PWM), is a method of transmitting information on a series of pulses



# Pulse Width Modulation

- ➊ Pulse Width Modulation (PWM), is a method of transmitting information on a series of pulses
- ➋ The data that is being transmitted is encoded on the width of these pulses to control the amount of power being sent to a load



# Pulse Width Modulation

- ➊ Pulse Width Modulation (PWM), is a method of transmitting information on a series of pulses
- ➋ The data that is being transmitted is encoded on the width of these pulses to control the amount of power being sent to a load
- ➌ Examples: Electric stoves, Lamp dimmers, and Robotic Servos



# Pulse Width Modulation

- ➊ Pulse Width Modulation (PWM), is a method of transmitting information on a series of pulses
- ➋ The data that is being transmitted is encoded on the width of these pulses to control the amount of power being sent to a load
- ➌ Examples: Electric stoves, Lamp dimmers, and Robotic Servos



# Pulse Width Modulation

- ➊ Pulse Width Modulation (PWM), is a method of transmitting information on a series of pulses
- ➋ The data that is being transmitted is encoded on the width of these pulses to control the amount of power being sent to a load
- ➌ Examples: Electric stoves, Lamp dimmers, and Robotic Servos

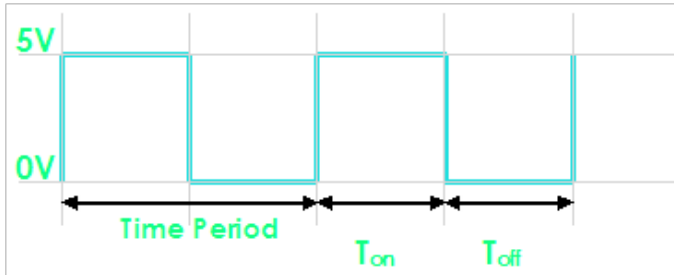




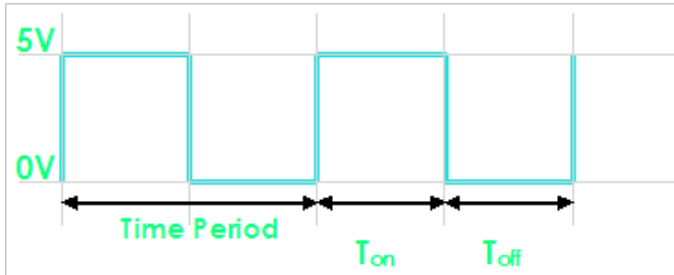
# Duty Cycle



# Duty Cycle



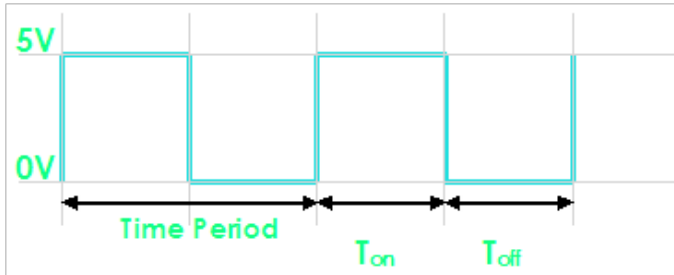
# Duty Cycle



- The signal remains "ON" for some time and "OFF" for some time.



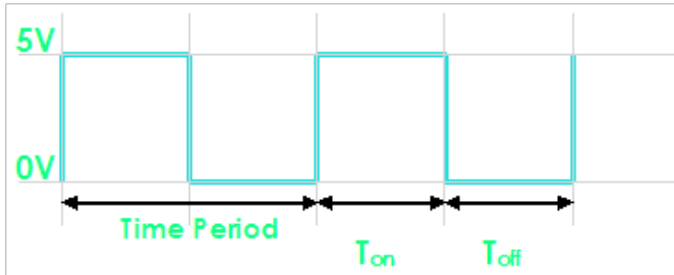
# Duty Cycle



- ✓ The signal remains "ON" for some time and "OFF" for some time.
- ✓  $T_{on}$  = Time the output remains high.



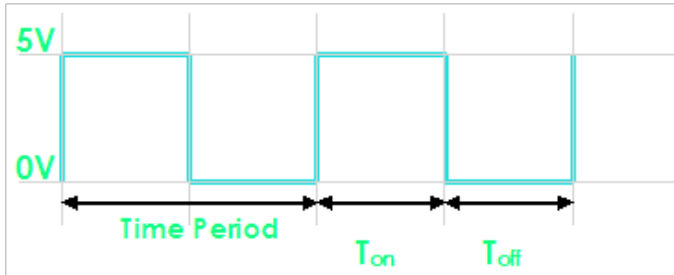
# Duty Cycle



- ✓ The signal remains "ON" for some time and "OFF" for some time.
- ✓  $T_{on}$  = Time the output remains high.
- ✓  $T_{off}$  = Time the output remains Low.



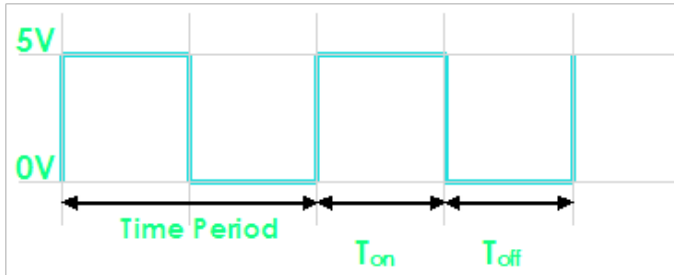
# Duty Cycle



- ✓ The signal remains "ON" for some time and "OFF" for some time.
- ✓  $T_{on}$  = Time the output remains high.
- ✓  $T_{off}$  = Time the output remains Low.
- ✓ When output is high the voltage is 5v



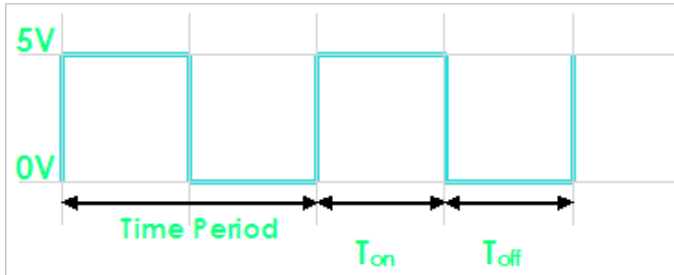
# Duty Cycle



- ✓ The signal remains "ON" for some time and "OFF" for some time.
- ✓  $T_{on}$  = Time the output remains high.
- ✓  $T_{off}$  = Time the output remains Low.
- ✓ When output is high the voltage is 5v
- When output is low the voltage is 0v



# Duty Cycle

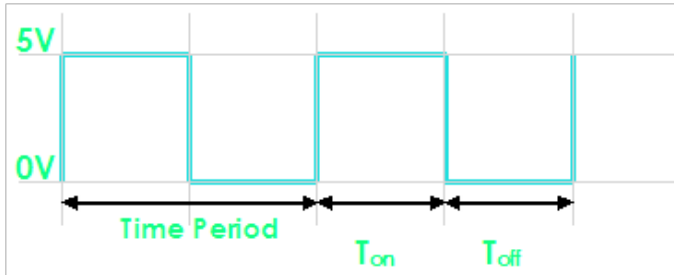


- ✓ The signal remains "ON" for some time and "OFF" for some time.
- ✓  $T_{on}$  = Time the output remains high.
- ✓  $T_{off}$  = Time the output remains Low.
- ✓ When output is high the voltage is 5v
- ✓ When output is low the voltage is 0v
- ✓ Time Period( $T$ ) =  $T_{on} + T_{off}$





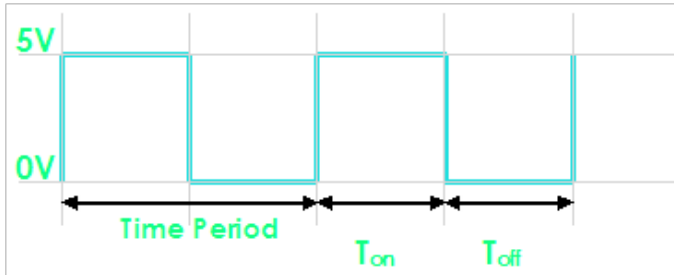
# Duty Cycle



- ✓ The signal remains "ON" for some time and "OFF" for some time.
- ✓  $T_{on}$  = Time the output remains high.
- ✓  $T_{off}$  = Time the output remains Low.
- ✓ When output is high the voltage is 5v
- ✓ When output is low the voltage is 0v
- ✓  $\text{Time Period}(T) = T_{on} + T_{off}$
- ✓  $\text{Duty Cycle} = T_{on} \times 100 / (T_{on} + T_{off})$



# Duty Cycle



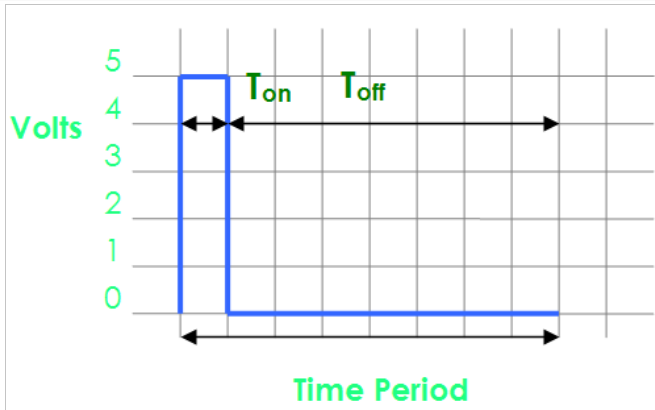
- ✓ The signal remains "ON" for some time and "OFF" for some time.
- ✓  $T_{on}$  = Time the output remains high.
- ✓  $T_{off}$  = Time the output remains Low.
- ✓ When output is high the voltage is 5v
- ✓ When output is low the voltage is 0v
- ✓  $\text{Time Period}(T) = T_{on} + T_{off}$
- ✓  $\text{Duty Cycle} = T_{on} \times 100 / (T_{on} + T_{off})$
- ✓ **Duty Cycle = 50%**



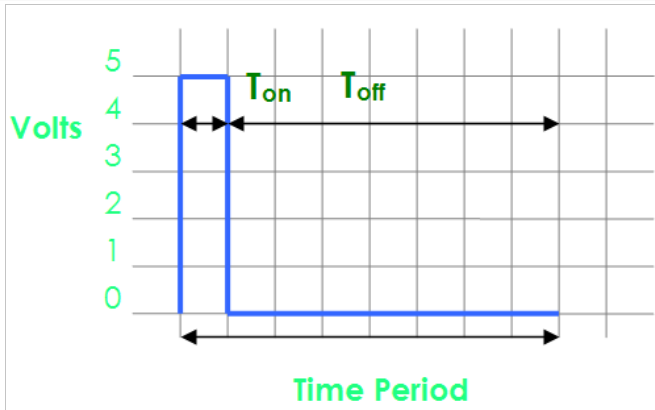
# Duty Cycle (Contd..)



## Duty Cycle (Contd..)



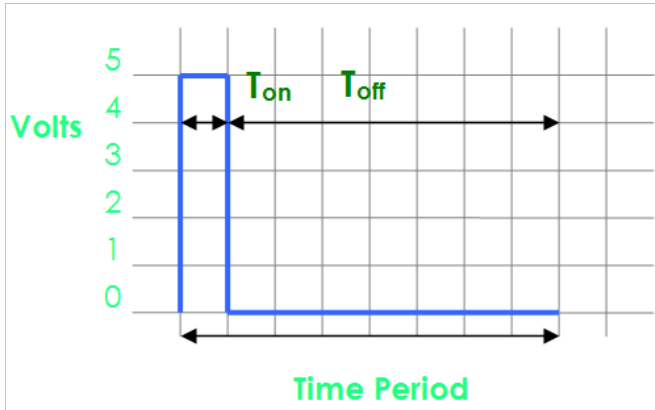
## Duty Cycle (Contd..)



- $T_{on}$  = Time the output remains high = 1



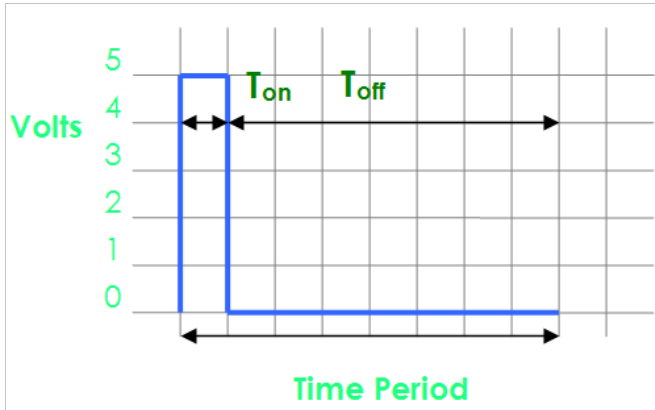
## Duty Cycle (Contd..)



- $T_{on}$  = Time the output remains high = 1
- $T_{off}$  = Time the output remains Low = 7



## Duty Cycle (Contd..)



- ✓  $T_{on}$  = Time the output remains high = 1
- ✓  $T_{off}$  = Time the output remains Low = 7
- ✓ Duty Cycle = 12.5%

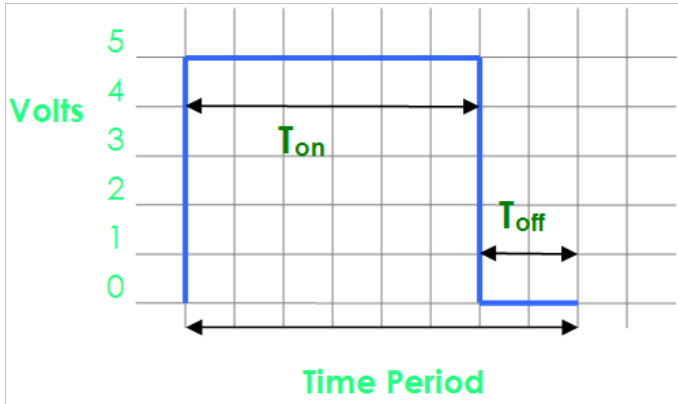


# Duty Cycle (Contd..)

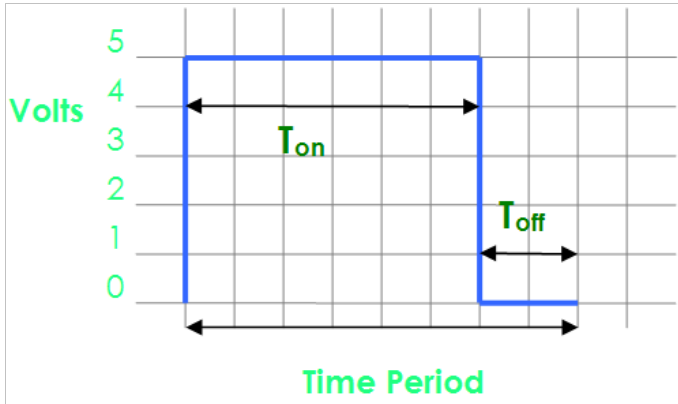




## Duty Cycle (Contd..)



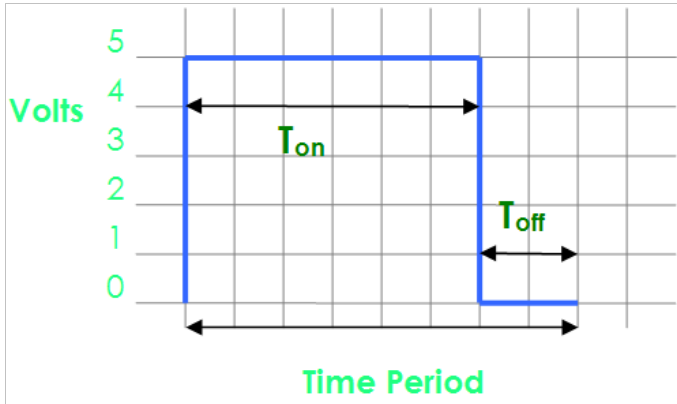
## Duty Cycle (Contd..)



●  $T_{on} = \text{Time the output remains High} = 6$



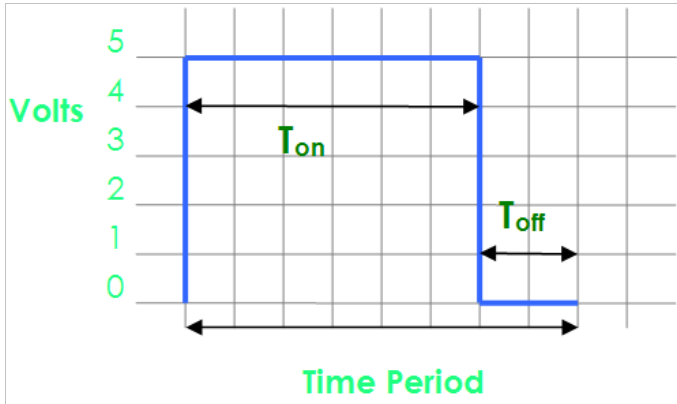
## Duty Cycle (Contd..)



- ✓  $T_{on}$  = Time the output remains High = 6
- ✓  $T_{off}$  = Time the output remains Low = 2



## Duty Cycle (Contd..)



- ✓  $T_{on}$  = Time the output remains High = 6
- ✓  $T_{off}$  = Time the output remains Low = 2
- ✓ Duty Cycle = 75%

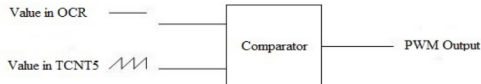


# PWM Generation in AVR



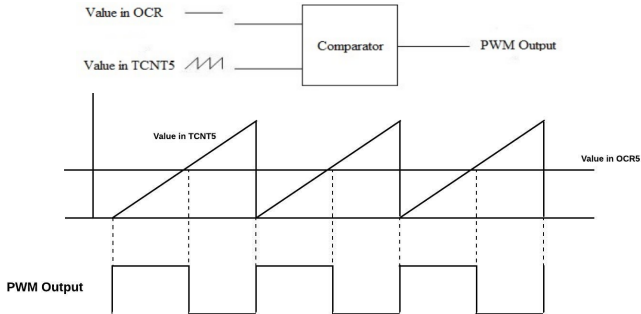
# PWM Generation in AVR

Pulse width waveform generated for motion control of Firebird V is:



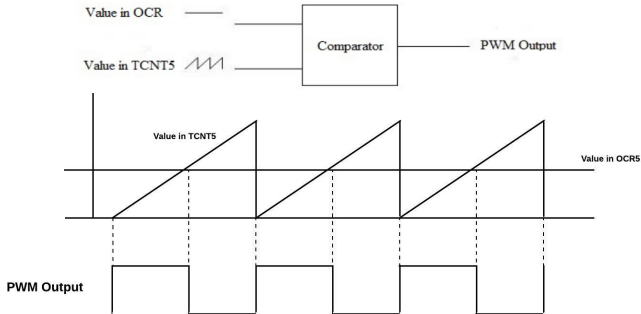
# PWM Generation in AVR

Pulse width waveform generated for motion control of Firebird V is:



# PWM Generation in AVR

Pulse width waveform generated for motion control of Firebird V is:



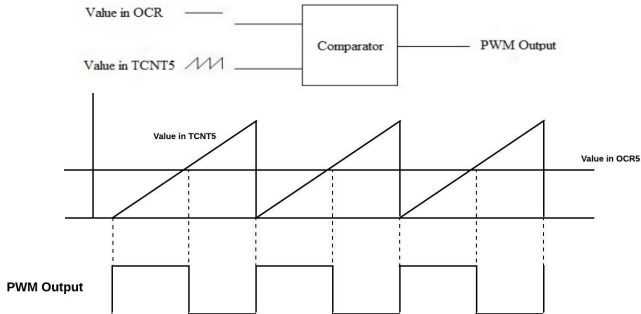
Its generation involves the use of following registers:





# PWM Generation in AVR

Pulse width waveform generated for motion control of Firebird V is:



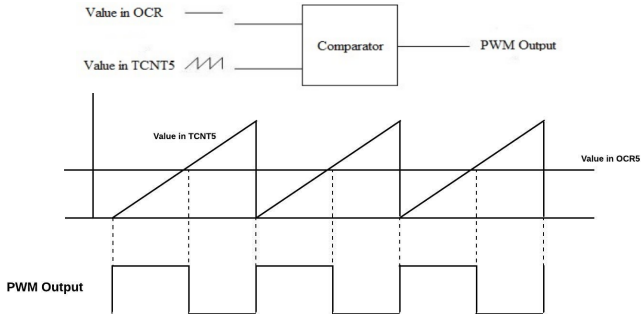
Its generation involves the use of following registers:

- 🔵 Timer/Counter register 5 (TCNT5)



# PWM Generation in AVR

Pulse width waveform generated for motion control of Firebird V is:



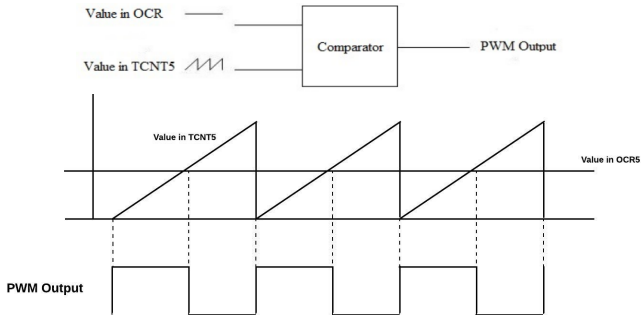
Its generation involves the use of following registers:

- ✓ Timer/Counter register 5 (TCNT5)
- ✓ Output Compare registers 5 (OCR5A and OCR5B)



# PWM Generation in AVR

Pulse width waveform generated for motion control of Firebird V is:



Its generation involves the use of following registers:

- ✓ Timer/Counter register 5 (TCNT5)
- ✓ Output Compare registers 5 (OCR5A and OCR5B)
- ✓ Timer/Counter Control registers (TCCR5A and TCCR5B)



# Timers in AVR



# Timers in AVR

- 1 The AVR microcontroller ATmega2560 has



# Timers in AVR

- 1 The AVR microcontroller ATmega2560 has
  - Two 8-bit timers (Timer 0 and Timer 2) and



# Timers in AVR

- ① The AVR microcontroller ATmega2560 has
  - Two 8-bit timers (Timer 0 and Timer 2) and
  - Four 16-bit timers (Timer 1, 3, 4 and 5)



# Timers in AVR

- ① The AVR microcontroller ATmega2560 has
  - Two 8-bit timers (Timer 0 and Timer 2) and
  - Four 16-bit timers (Timer 1, 3, 4 and 5)
- ② For speed control of Firebird V, Timer 5 is used.





# Timer/Counter 5 (TCNT5)



# Timer/Counter 5 (TCNT5)

- 1 The Timer/Counter is a register that increments its value after every clock cycle.



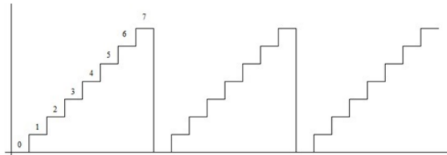
# Timer/Counter 5 (TCNT5)

- 1 The Timer/Counter is a register that increments its value after every clock cycle.
- 2 The maximum value depends upon the resolution of Counter.



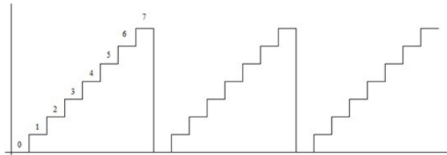
# Timer/Counter 5 (TCNT5)

- 1 The Timer/Counter is a register that increments its value after every clock cycle.
- 2 The maximum value depends upon the resolution of Counter.
- 3 For example, a 3 bit counter will have 8 values (i.e. 0-7). Its waveform will be seen as follows:



# Timer/Counter 5 (TCNT5)

- ❶ The Timer/Counter is a register that increments its value after every clock cycle.
- ❷ The maximum value depends upon the resolution of Counter.
- ❸ For example, a 3 bit counter will have 8 values (i.e. 0-7). Its waveform will be seen as follows:

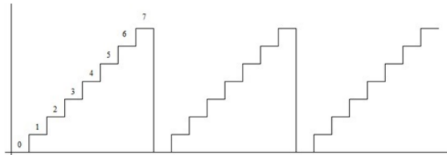


- ❹ For n-bit counter, maximum value =  $2^n - 1$ .



# Timer/Counter 5 (TCNT5)

- 1 The Timer/Counter is a register that increments its value after every clock cycle.
- 2 The maximum value depends upon the resolution of Counter.
- 3 For example, a 3 bit counter will have 8 values (i.e. 0-7). Its waveform will be seen as follows:

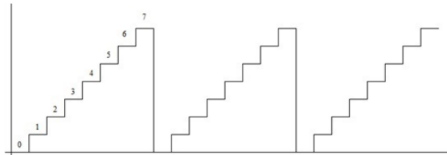


- 4 For n-bit counter, maximum value =  $2^n - 1$ .
- 5 The Timer/Counter 5 is a 16 bit register.



# Timer/Counter 5 (TCNT5)

- 1 The Timer/Counter is a register that increments its value after every clock cycle.
- 2 The maximum value depends upon the resolution of Counter.
- 3 For example, a 3 bit counter will have 8 values (i.e. 0-7). Its waveform will be seen as follows:



- 4 For n-bit counter, maximum value =  $2^n - 1$ .
- 5 The Timer/Counter 5 is a 16 bit register.
- 6 We use it in 8-bit mode, for PWM generation.



# Output Compare Register (OCR5A, OCR5B and OCR5C)





# Output Compare Register (OCR5A, OCR5B and OCR5C)

- 1 The value of the Timer/Counter 5 is constantly compared with a reference value.



# Output Compare Register (OCR5A, OCR5B and OCR5C)

- 1 The value of the Timer/Counter 5 is constantly compared with a reference value.
- 2 This reference value is given in the Output Compare Register (OCR).



# Output Compare Register (OCR5A, OCR5B and OCR5C)

- 1 The value of the Timer/Counter 5 is constantly compared with a reference value.
- 2 This reference value is given in the Output Compare Register (OCR).
- 3 Output Compare Registers associated with Timer 5 for PWM generation: OCR5A, OCR5B and OCR5C.



# Output Compare Register (OCR5A, OCR5B and OCR5C)

- 1 The value of the Timer/Counter 5 is constantly compared with a reference value.
- 2 This reference value is given in the Output Compare Register (OCR).
- 3 Output Compare Registers associated with Timer 5 for PWM generation: OCR5A, OCR5B and OCR5C.
- 4 For motion control of Firebird V, we use OCR5A and OCR5B registers.



# Output Compare Register (OCR5A, OCR5B and OCR5C)

- 1 The value of the Timer/Counter 5 is constantly compared with a reference value.
- 2 This reference value is given in the Output Compare Register (OCR).
- 3 Output Compare Registers associated with Timer 5 for PWM generation: OCR5A, OCR5B and OCR5C.
- 4 For motion control of Firebird V, we use OCR5A and OCR5B registers.
- 5 OCR5A is associated with the OC5A pin (PORTL3). This pin is connected to the enable(EN2) pin of motor driver, which is associated with the left motor.



# Output Compare Register (OCR5A, OCR5B and OCR5C)

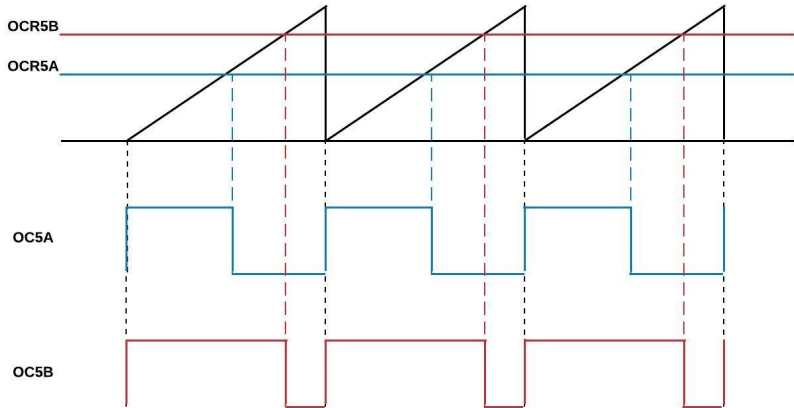
- 1 The value of the Timer/Counter 5 is constantly compared with a reference value.
- 2 This reference value is given in the Output Compare Register (OCR).
- 3 Output Compare Registers associated with Timer 5 for PWM generation: OCR5A, OCR5B and OCR5C.
- 4 For motion control of Firebird V, we use OCR5A and OCR5B registers.
- 5 OCR5A is associated with the OC5A pin (PORTL3). This pin is connected to the enable(EN2) pin of motor driver, which is associated with the left motor.
- 6 Similarly, OCR5B is associated with the OC5B pin (PORTL4). This pin is connected to the enable(EN1) pin of motor driver, which is associated with the right motor.



# PWM signal for Left and Right motor



# PWM signal for Left and Right motor





# TCCR5A- Timer Counter Control Register A



# TCCR5A- Timer Counter Control Register A

Bit	Symbol	Description	Bit Value
7	COM5A1	Compare Output Mode for Channel A bit 1	1
6	COM5A0	Compare Output Mode for Channel A bit 0	0
5	COM5B1	Compare Output Mode for Channel B bit 1	1
4	COM5B0	Compare Output Mode for Channel B bit 0	0
3	COM5C1	Compare Output Mode for Channel C bit 1	1
2	COM5C0	Compare Output Mode for Channel C bit 0	0
1	WGM11	Waveform Generation Mode bit 1	0
0	WGM10	Waveform Generation Mode bit 0	1



# TCCR5A- Timer Counter Control Register A

Bit	Symbol	Description	Bit Value
7	COM5A1	Compare Output Mode for Channel A bit 1	1
6	COM5A0	Compare Output Mode for Channel A bit 0	0
5	COM5B1	Compare Output Mode for Channel B bit 1	1
4	COM5B0	Compare Output Mode for Channel B bit 0	0
3	COM5C1	Compare Output Mode for Channel C bit 1	1
2	COM5C0	Compare Output Mode for Channel C bit 0	0
1	WGM11	Waveform Generation Mode bit 1	0
0	WGM10	Waveform Generation Mode bit 0	1

- ① There are 2 types of bits in TCCR5A: Compare output mode bit  
waveform generation mode bit.



# TCCR5A- Timer Counter Control Register A

Bit	Symbol	Description	Bit Value
7	COM5A1	Compare Output Mode for Channel A bit 1	1
6	COM5A0	Compare Output Mode for Channel A bit 0	0
5	COM5B1	Compare Output Mode for Channel B bit 1	1
4	COM5B0	Compare Output Mode for Channel B bit 0	0
3	COM5C1	Compare Output Mode for Channel C bit 1	1
2	COM5C0	Compare Output Mode for Channel C bit 0	0
1	WGM11	Waveform Generation Mode bit 1	0
0	WGM10	Waveform Generation Mode bit 0	1

- There are 2 types of bits in TCCR5A: Compare output mode bit waveform generation mode bit.
- Compare Output Mode bits decide the action to be taken when counter(TCNT5) value matches reference value in Output Compare Register(OCR5).



# Compare Output Mode bits



# Compare Output Mode bits

**Table 17-4.** Compare Output Mode, Fast PWM

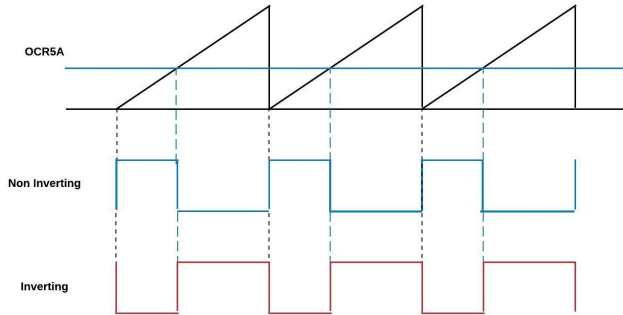
COMnA1 COMnB1 COMnC1	COMnA0 COMnB0 COMnC0	Description
0	0	Normal port operation, OCnA/OCnB/OCnC disconnected.
0	1	WGM13:0 = 14 or 15: Toggle OC1A on Compare Match, OC1B and OC1C disconnected (normal port operation). For all other WGM1 settings, normal port operation, OC1A/OC1B/OC1C disconnected.
1	0	Clear OCnA/OCnB/OCnC on compare match, set OCnA/OCnB/OCnC at BOTTOM (non-inverting mode).
1	1	Set OCnA/OCnB/OCnC on compare match, clear OCnA/OCnB/OCnC at BOTTOM (inverting mode).



# Cont..



## Cont..

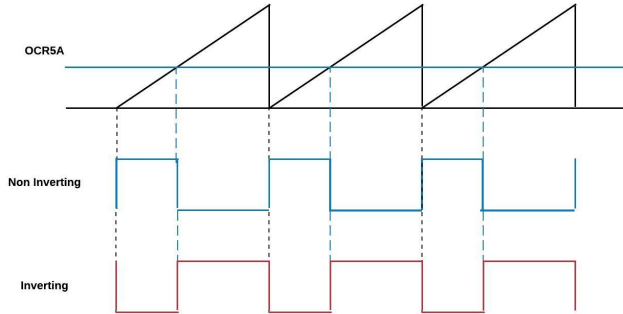


- ① We are using non-inverting mode for PWM generation.





## Cont..



- ① We are using non-inverting mode for PWM generation.
- ② Non-inverting mode and inverting mode



# Waveform Generation Bit



# Waveform Generation Bit

**Table 17-2.** Waveform Generation Mode Bit Description<sup>(1)</sup>

Mode	WGMn3	WGMn2 (CTCn)	WGMn1 (PWMn1)	WGMn0 (PWMn0)	Timer/Counter Mode of Operation	TOP	Update of OCRnX at	TOVn Flag Set on
0	0	0	0	0	Normal	0xFFFF	Immediate	MAX
1	0	0	0	1	PWM, Phase Correct, 8-bit	0x00FF	TOP	BOTTOM
2	0	0	1	0	PWM, Phase Correct, 9-bit	0x01FF	TOP	BOTTOM
3	0	0	1	1	PWM, Phase Correct, 10-bit	0x03FF	TOP	BOTTOM
4	0	1	0	0	CTC	OCRnA	Immediate	MAX
5	0	1	0	1	Fast PWM, 8-bit	0x00FF	BOTTOM	TOP
6	0	1	1	0	Fast PWM, 9-bit	0x01FF	BOTTOM	TOP
7	0	1	1	1	Fast PWM, 10-bit	0x03FF	BOTTOM	TOP
8	1	0	0	0	PWM, Phase and Frequency Correct	ICRn	BOTTOM	BOTTOM
9	1	0	0	1	PWM, Phase and Frequency Correct	OCRnA	BOTTOM	BOTTOM
10	1	0	1	0	PWM, Phase Correct	ICRn	TOP	BOTTOM
11	1	0	1	1	PWM, Phase Correct	OCRnA	TOP	BOTTOM
12	1	1	0	0	CTC	ICRn	Immediate	MAX
13	1	1	0	1	(Reserved)	–	–	–
14	1	1	1	0	Fast PWM	ICRn	BOTTOM	TOP
15	1	1	1	1	Fast PWM	OCRnA	BOTTOM	TOP



# TCCR5B- Timer Counter Control Register B



# TCCR5B- Timer Counter Control Register B

Bit	Symbol	Description	Bit Value
7	ICNC5	Input Capture Noise Canceller	0
6	ICES5	Input Capture Edge Select	0
5	–	Reserved Bit	0
4	WGM53	Waveform Generation Mode bit 3	0
3	WGM52	Waveform Generation Mode bit 2	1
2	CS52	Clock Select	0
1	CS51	Clock Select	1
0	CS50	Clock Select	1



# TCCR5B- Timer Counter Control Register B

Bit	Symbol	Description	Bit Value
7	ICNC5	Input Capture Noise Canceller	0
6	ICES5	Input Capture Edge Select	0
5	–	Reserved Bit	0
4	WGM53	Waveform Generation Mode bit 3	0
3	WGM52	Waveform Generation Mode bit 2	1
2	CS52	Clock Select	0
1	CS51	Clock Select	1
0	CS50	Clock Select	1



# Clock Select Bits



# Clock Select Bits

**Table 17-6.** Clock Select Bit Description

CSn2	CSn1	CSn0	Description
0	0	0	No clock source. (Timer/Counter stopped)
0	0	1	$clk_{I/O}/1$ (No prescaling)
0	1	0	$clk_{I/O}/8$ (From prescaler)
0	1	1	$clk_{I/O}/64$ (From prescaler)
1	0	0	$clk_{I/O}/256$ (From prescaler)
1	0	1	$clk_{I/O}/1024$ (From prescaler)
1	1	0	External clock source on Tn pin. Clock on falling edge
1	1	1	External clock source on Tn pin. Clock on rising edge

$$PWM_{frequency} = Clock_{frequency} / (N * 255)$$

where,

$$Clock_{frequency} = 14745600Hz$$

N = prescaler factor





# Summary



# Summary

In order to use Fast PWM mode to control the speed of dc motors of Firebird V. We have to initialize following registers with the corresponding values:



# Summary

In order to use Fast PWM mode to control the speed of dc motors of Firebird V. We have to initialize following registers with the corresponding values:

✔  $TCNT5L = 0x00$



# Summary

In order to use Fast PWM mode to control the speed of dc motors of Firebird V. We have to initialize following registers with the corresponding values:

- ✓ TCNT5L = 0x00
- ✓ TCCR5A = 0xA9



# Summary

In order to use Fast PWM mode to control the speed of dc motors of Firebird V. We have to initialize following registers with the corresponding values:

- ✓ TCNT5L = 0x00
- ✓ TCCR5A = 0xA9
- ✓ TCCR5B = 0x0B



# Summary

In order to use Fast PWM mode to control the speed of dc motors of Firebird V. We have to initialize following registers with the corresponding values:

- ✓ TCNT5L = 0x00
- ✓ TCCR5A = 0xA9
- ✓ TCCR5B = 0x0B
- ✓ OCR5AH = 0x00



# Summary

In order to use Fast PWM mode to control the speed of dc motors of Firebird V. We have to initialize following registers with the corresponding values:

- ✓ TCNT5L = 0x00
- ✓ TCCR5A = 0xA9
- ✓ TCCR5B = 0x0B
- ✓ OCR5AH = 0x00
- ✓ OCR5AL = 0xFF



# Summary

In order to use Fast PWM mode to control the speed of dc motors of Firebird V. We have to initialize following registers with the corresponding values:

- ✓ TCNT5L = 0x00
- ✓ TCCR5A = 0xA9
- ✓ TCCR5B = 0x0B
- ✓ OCR5AH = 0x00
- ✓ OCR5AL = 0xFF
- ✓ OCR5BH = 0x00





# Summary

In order to use Fast PWM mode to control the speed of dc motors of Firebird V. We have to initialize following registers with the corresponding values:

- ✓ TCNT5L = 0x00
- ✓ TCCR5A = 0xA9
- ✓ TCCR5B = 0x0B
- ✓ OCR5AH = 0x00
- ✓ OCR5AL = 0xFF
- ✓ OCR5BH = 0x00
- ✓ OCR5BL = 0xFF



# Thank You!

